FUNCTIONAL SPECIFICATION OF ECONET-X25 GATEWAY

CONTENTS

1 Introduction

2 Selection of Gateway Protocol Converter

3 Basic Gateway Function

4 Effect of Protocol Conversions on Basic Gateway Function

4.1 Yellow Book Transport Service

4:2 X29

5 Configuration, Monitoring and Management

1 Introduction

The Econet-X25 gateway is a specially built unit which fits between the Econet and an X25 network to allow transfer of data between the two. Any Econet user wishing to access the external X25 network will require a network service interface. The gateway will receive network service(NS) connection requests over the Econet and forward them over the X25 network using the Symicron Telematics Software(STS) X25 interface to an X25 server. Similarly, incoming calls from the X25 network will be passed from the STS module to the NS module and thence to an Econet server. The gateway software between the NS and STS modules will have to perform interface conversion plus protocol conversion where appropriate.

Any call through the gateway will have some protocol conversion applied to it. Initially the gateway will support just three such conversions. The first is effectively X25 over NS to X25 over STS: This allows an NS user to make X25 calls to remote X25 servers, but restricts his repertoire of commands to whatever the NS will support (e.g. the Q-bit is not supported by the NS). This conversion may also be used by remote X25 users calling an Econet NS server, but is subject to restricted destination addressing.

The second protocol conversion is Yellow Book Transport Service over NS to Yellow Book Transport Service over X25. Use of the transport service over the gateway will ensure maximum addressing flexibility. The transport service user is minimally restricted by the gateway: the Address primitive will not be supported, and the Connect primitive should contain no more bytes than will fit in a fast select packet (128).

The third protocol conversion is X29 over NS to X29 over X25. X29 over NS looks a little like TS29 in that a leading byte is required in each complete data unit to indicate whether the data is control (e.g set PAD parameters) or not. There does not seem to be any need to support true TS29 with each record preceded by a length byte, since the NS has the concept of data unit size. This protocol permits an Econet user to act as a PAD and access remote X29 hosts. If he wants to access a TS29 host, this is covered by the YBTS protocol conversion.

In general the gateway should not need to police the X25/Network service protocol. This will be done by the STS module on one side and the NS module on the other.

The gateway should not often need to introduce additional STS primitives or NS items i:e the primitives received from one side can usually be passed on to the other side with a minimal amount of header translation. Similarly, there will be relatively few occasions when primitives received from the STS or NS modules are discarded without being forwarded. Some exceptions to this are:

(a) Whenever a connect primitive is received with an unknown destination (called address), the gateway will have the responsibility of disconnecting the call. In this case the arriving primitive is discarded and a different primitive issued in the reverse direction.

(b) Support of yellow book transport service(YBTS) will require that the gateway turn a NS Connect Confirm into an STS Accept primitive plus an STS data primitive. Similarly, an Accept primitive from the STS module will be discarded and the subsequent Data primitive used as the basis of a Connect Response to the NS.

(c) Reset primitives do not directly map, since STS and NS do not have the same Resetting procedures.

(d) Splitting and blocking of data items will be required to match the interfaces on each side of the gateway. So Data primitives received from one side will cause Data primitives to be issued to the other side, but there will not be a non-to-one correspondence. There will also be implementation specific flow control procedures for both X25 and the Econet NS, which will have to be appropriately used. These procedures will not be symmetric across the gateway.

There should be no need for the gateway to accept an X25 call before forwarding the call request/connect: this would be in violation of note 3. This precludes the use of transport service calls whose connect information will not be completely contained in a call request packet or an NS connect item. The support of fast select is necessary, up to the full 128 bytes, with a limited facilities field of, say, 20 bytes. This may have an impact on the X25 porting. On the Econet network service side, a similar amount of 'facilities' and 'data' will be needed.

The gateway software resides entirely within the special Z80 board. This is a two processor board with one processor given over to the STS module and the other given over to the gateway function. Access to the network service interface in the 6502 is made particularly easy by using the Tube (see appendix J of BBC CP/M manual). The OSWORD calls by which the network service is used are accesible directly from the standard BBC Z80 second processor, and it is not foreseen that there will be any problem including the relevant Z80 code in the gateway processor. All the usual calls (OSBYTE, OSFILE etc) are available across the tube which makes access to files straightforward:

2 Selection of Gateway Protocol Converter

This is done by inspection of the protocol ID in the received Connect primitive. This is a four byte field at the start of user data in the STS primitive, or a separate network protocol identifier (NPI) in the Connect indication received from the NS.

The three conversions identified currently are:

(a) X25 over NS < - - - > standard X25
(b) YBTS over NS < - - > YBTS over X25
(c) X29 over NS < - - - > X29 over X25 (probably outbound calls only)

From the STS module, the values to look out for are:

Hex'01xxxxxx' -.incoming call from X25 is a PAD call to an X29 host on the Econet. There is not necessarily an immediate requirement for this type of call, but it would be useful to call an X29 gateway configuration server within the gateway one day.

Two possibilities based on called subaddress:

00 - no protocol conversion required - call to gateway server (not implemented initially)
01 - X29 to X29 over NS Dialogue with caller to ascertain TS address

Hex'7FFFFFFF' - incoming call from X25 is supporting YBTS. Could be used for file transfer or TS29 call to Econet host: the gateway does not need to know. Subaddress is not important.

Any other value - incoming call from X25 has no higher protocol identified

The NS currently allows just one byte for the NPI, but could use more (4?) The assignment of protocol ID's is yet to be made. For the time being let us assume the same values as the STS will present:

So from the NS, the values to look out for are:

Hex'01xxxxxx' - incoming call from NS is an X29 PAD call to an X29 host
Hex'7FFFFFFF' - incoming call from NS is supporting YBTS
Any other value - incoming call from NS has no higher protocol identified

3 Basic Gateway Function

The minimum requirement involves mapping messages from STS into messages to the NS and vice-versa. There will be a need to accept data from each side in the appropriate type of buffer and pass it to the other side in whatever type of buffer it accepts. This will mean copying of data and translation of headers etc. Because each side can exert flow control back to the gateway, there will be a need to buffer data occasionally, on a per call basis.

Actions on receiving items from the STS module/NS:

STS Connect Primitive

The protocol ID will be inspected for the X29 ID or the yellow book ID. If either of these is found, proceed as described in later sections. Otherwise, abstract the sub-address, convert to an Econet address using a simple 'title-to-address' table, and build an appropriate Connect Request OSWORD call using a free Econet call 'handle'. The called address may or may not look like the incoming called address, but the calling address should still identify the remote X25 user. Facilities from the STS Connect (Fast Select, Reverse Charge etc) will in some circumstances need to be passed on. The mechanism for doing this has to be decided: non-standard Quality of Service, or some other means.

There is a possibility that the call cannot be passed on for lack of a valid destination. In this case, the call will be rejected by the gateway software issuing a Disconnect Primitive back to the STS module. This is one case where the gateway becomes a transient end-point of a call and needs to be aware of the 'state' of the connection. In fact, the gateway's Disconnect completes the closedown, and no reply is expected from the STS module.

NS Connect Indication

Once the gateway has started to read in this primitive, it will read in the whole of it. After all, it will all be within the 6502 at this time and is not going to exceed 200 bytes. The network protocol addressing information will contain the X25 address to be called, or conceivably a title which can be used to derive such an address. The network protocol identifier should contain a value which can select any of the following:

Straight X25 - NPAI address in X25 form

Yellow Book Transport Service - NPAI is equivalent to a YB address (This includes TS29 calls without gateway conversion)

TS29 with TS29 to X29 conversion in the gateway

It is possible that the gateway is unable to match the called address received from the NS in the address translation tables appropriate to the desired protocol. If so, the gateway will need to issue a Disconnect Request back to the network service to reject the call. No further indication or response will be received from the NS for this call.

STS Accept

This follows a Connect Request from the Econet to X25. There may be some address conversion if extended formats are used. Facilities, if present, will need some simple translation. STS may return a channel number besides the call identifier but this is not particularly useful to us, (though we may need to enter it in future requests relating to this call). The Accept primitive is passed on to the NS as a Connect Response.

NS Connect Confirm

This follows a Connect primitive from X25 to the Econet. There may be some address translations involved depending on the gateway protocol conversion in use on the current channel. The Connect Confirm is passed on to the STS module as an Accept primitive.

STS Disconnect

All buffered data items, in both directions of the call, should be discarded and flags reset to channel out-of-use as necessary. The Disconnect primitive will be passed on to the NS as a Disconnect Request with appropriate cause etc (there will be a need to translate the reason from X25 to NS codes). Note that this primitive appears to completely break the connection between the gateway module and STS module, and requires no further confirmation from the gateway except, apparently, when cause Hex'FF' is received from the STS module. This cause indicates a disconnect due to an error by the user, and should be confirmed by the gateway via a Disconnect primitive back to the STS module.

p.s. Check with Symicron that this is the case: that a Disconnect "confirm" is required if and only if the Disconnect primitive received from the STS module has cause Hex'FF'. (Is this an equivalent of a diagnostic packet?)

Also check whether an incoming clear from X25 is destructive of data ahead of it that is yet to be picked up by the gateway: apparently it is not.

Also check apparent STS manual error:
Page 6 type = 4 Fast Disconnect primitive
type = 5 Disconnect primitive (implied push)

Page 19/20 type = 4 Abort Disconnect
type = 5 Normal Disconnect

Will STS accept addresses+facilities+data to a total byte count > 126 if we specify larger buffers?
Will STS allow user data in clear packets following fast select call set-up? (CCITT 1980 for further study, PSS not allowed, CCITT 1984 allowed)

NS Disconnect Indication

This causes all buffered data for the channel (in both directions) to be discarded and flags reset etc to indicate that the call is no longer in existence. The Disconnect is passed to the STS module as a Disconnect primitive with reason translation as necessary. No confirmation is subsequently expected from the STS module. Note that NS user data may not be passed on to the STS if the original call was set up as non-fast select. If it was, there is still a decision to be made as to whether to attempt a fast select clear after the call has been established. (see note above). Again, no disconnect "confirmation" is required from the gateway:

p.s. It seems possible that the STS module may still have outstanding primitives waiting to be read by the gateway. If so, these should be ignored after being read until a new call is set up on the same channel or signature. Check with Symicron whether this can happen.

STS Reset

All buffered data items in the STS to Econet direction should be discarded and flags set as necessary. The reset is passed on as a Reset request to the NS. There will be a translation on the reset cause etc. Note that a Reset in this direction should be confirmed by a Reset primitive back to the STS. We can assume end-to-end operation for such resets since the NS will have to return a Reset Confirm in due course: Therefore we do not need to inject a Reset primitive back to the STS. Note that this is not the case in the reverse direction: i.e the STS does not return a Reset primitive following a Reset sent from the gateway.

NS Reset Indication

All buffered data items, in both directions, should be discarded and flags set as necessary. The Reset Indication is passed on (with reason code translation) to the STS module as a Reset primitive. As the STS appears not to confirm the Reset, the gateway must inject a Reset Response back to the NS.

p:s: confirm this assymetric STS behaviour with Symicron

NS Reset Confirmation

Already alluded to, this follows the sending of a Reset Request to the NS: The Reset Confirmation will be passed on to the STS as a Reset Primitive.

STS Expedited

This will be passed to the NS as an expedited data request. It may be sent even if the NS has blocked the current data flow and the last data primitive sent to the NS indicated ' not end of data unit'. There will be just one byte of expedited data. There is no confirmation by the NS and none expected by the STS. If expedited data is sent too often when the data flow is blocked, the expedited data will simply be lost.

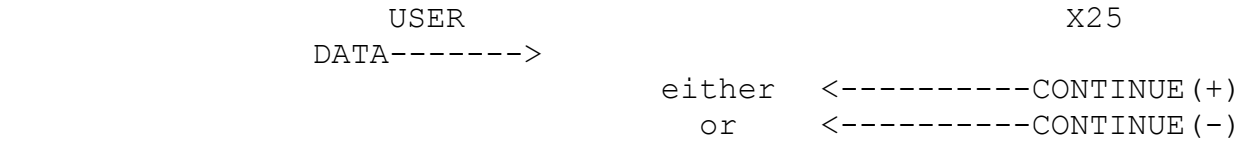
NS Expedited

This may be up to 32 bytes in length. If more than one byte is sent, the rest will be discarded and the single byte passed to the STS via an Expedited primitive. No confirmation will be received from the STS and if this Expedited primitive follows before the STS has received an interrupt confirm, the data will be lost.

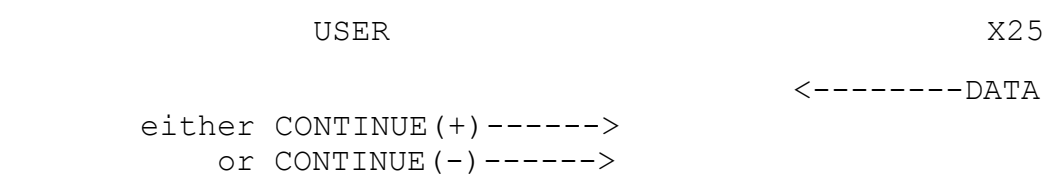
Data

By and large, data received from either the NS or STS will be passed unaltered, but perhaps in different sized chunks. There will be an additional byte at the start of each data item passed to/from the NS to map from/to the Q-bit indicator in the STS interface, when certain protocol conversions are used (see appendix):

Principles governing data operations over the STS interface



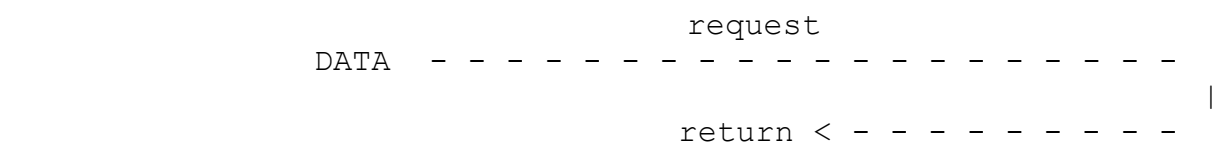
There will always be one such message from the X25 indicating whether the X25 module can take more data. Remember certain rules apply to the size of the data: if the 'more data' bit is set the number of bytes sent must be a multiple of 128 (the packet size).



Note that the CONTINUE(-) command will cause the X25 module to issue a level 3 RNR to the remote end. This is a rather active way of stopping the flow. The CONTINUE(+) command can be issued at any time and will cause an RR to be sent to the remote end of the X25 line if it follows a CONTINUE(-) from the user. The same rules apply to data size as in the USER->X25 direction: the X25 module will deliver up chunks of data in arbitrary sizes unless the 'more data' bit is set in which case the chunk will be a multiple of 128 bytes (i.e corresponds to one or more complete packets)

Principles governing data operations over the NS interface

These are OSWORD calls by the user



Note that (1) the reply is immediate (routine call)
(2) the amount of data is restricted to around 120 bytes
(3) the data is copied to an Econet NS buffer before returning, so that the user can immediately re-use the buffer passed in the request
(4) the parameter block passed in the return will indicate whether the user can send more data or not. There will be a little slack so as to permit a full data request when the blocked indicator has just turned true

If the user wishes to establish whether he can send data after being 'blocked' by flow control from the far end, he should inspect the parameter block returned in the reply to a status request.

If the user wishes to establish whether there is data to pick up from the Econet NS, he can do this by inspecting the parameter block returned in reply to a status request. Failure to issue a 'get data' request to the Econet NS will result in the remote NS user being blocked if the data awaiting delivery to the local user exceeds a certain amount.

Intermediate buffering within the simple gateway is necessary for the following reasons:

(1) The X25 and Econet NS have different effective windows for the same call.

(2) Data units received from the X25 or Econet NS will not necessarily be of a suitable size to pass to the other service. This particularly applies to data to be sent to the X25 module with the 'more data' bit set.

(3) Overall buffering limitations (i.e across all connections) in both X25 and Econet NS modules may cause override normal windowing within the modules.

Data indication from Econet NS

The gateway software will not receive data units from the Econet NS until it specifically requests data. Prior to issuing such a request, the gateway should ensure that (a) the X25 port is not blocked, and (b) the Econet NS actually has some data to pass on. Nevertheless, once the data has been 'collected' from Econet NS, there is no guarantee that it can be passed immediately to the X25 module. This is primarily due to the fact that the data unit will not necessarily be of the correct size to hand on.
Consequently, there will often be a 128 byte buffer in the ' partially filled' state, held by the gateway, waiting to be passed to the X25 module when filled. There should not, however, be a need to buffer more than 128 bytes of data in this direction.

Data primitive from STS

The data-units from the X25 module are unsolicited and may be received at too fast a rate to for the remote user of the Econet NS to handle. If so, the gateway, which is aware of outbound data backing up inside the local end of the Econet NS, will have to be prepared to buffer up data locally until the Econet clears the blocked data. In addition, each item received from the X25 is probably too large to pass on to the Econet NS without breaking up into smaller chunks.

Channel data required per call

STS signature	1 byte
Econet handle	1 byte
STS->NS data queue:	
Head of queue	2 bytes
Tail of queue (possibly)	2 bytes
(n.b. item on queue contains data, pointer to start of this data, length of data, M and Q bit settings, pointer to next item)	
NS->STS partial data item	2 bytes
(n.b item as defined above)	
Status flags	2 bytes
Protocol conversion in use	
Blocked in STS direction	
Blocked in Econet direction	
Flag to show if channel in use by call	

4 Effect of Protocol Conversions on Basic Gateway Operations

One very important difference between X25 and the NS is the lack of a Q-bit equivalent in the NS: The question arises as to what will be done about data packets received from the X25 line with the Q-bit set. In fact, this will be dependent on the converter selected. For X25 over NS, we restrict the user to not using the Q-bit: For the other two converters the initial byte in each NS data unit will map into the Q-bit of the corresponding X25 data packet sequence.

X25 over NS

NS Connect to STS Connect

Called address taken from NPAI
Facilities built up from QOS
Fast select invoked if user data more than 12 bytes

STS Connect to NS Connect

Called sub-address used to access look-up table of Econet network addresses and NPAI built up
QOS built up from facilities

NS Connect Confirm to STS Accept

Facilities built up from QOS

STS Accept to NS Connect Response

QOS built up from facilities
Called address used to update NPAI

NS data to STS data

The Q-bit is left unset in all data primitives sent to the STS module.

STS data to NS data

The Q-bit is ignored in data primitives received from the STS module.

4.1 YBTS over NS

NS Connect to STS Connect

NPAI contains called and calling YB addresses. These may include titles. Title to address mapping will be performed on the first component of the called address if not an X25 address. Fast select will be invoked when calling STS if YB connect primitive cannot be encoded in a normal call request packet. YB Called and calling addresses Will be

appropriately entered in primitive

STS Connect to NS Connect

YB called address may start with a title which can be used to identify a NS subscriber after suitable translation in a look-up table. One table could probably be used for both inbound and outbound title to address mappings.

NS Connect Confirm to STS Accept + STS Data

YB addresses will be placed in the NPAI field. It is believed that the X25 accept packet should not be used for TS Accept information (check). Therefore this Connect Confirm primitive will be split into two consecutive primitives, one STS Connect and one STS Data, when passed on.

STS Accept + STS Data to NS Connect Response

The initial STS Accept should be discarded, and the next received STS Data primitive used as the basis of the NS Connect Response. This means that the gateway needs to record the state of the channel as 'awaiting YB Accept from STS'

NS Data to STS Data

The first byte of NS data should be used to set the Q-bit

STS Data to NS Data

The Q-bit should be used to set the first byte of NS data

4.2 X29 over NS

There is no need fully convert to TS29 here: the NS has a concept of record size, unlike YBTS.

NS Connect to STS Connect

The NPAI should be used to select the outbound X25 address, possibly requiring access to the title to address translation table. The X25 protocol ID will come from the user data rather than the NPI.

STS Accept to NS Connect Response

As normal

NS Data to STS Data

The first byte should be used to set the Q-bit

STS Data to NS Data

The Q-bit should be used to set up the first byte

5 Configuration, Monitoring and Management

5.1 Configuration

There are two configuration tasks that need to be done. The first is the configuration of the STS module, and the second is the configuration of the gateway tables. The STS module needs the following information at system start-up in order to operate the X25 line correctly:

Logical Channels Ranges
Timeouts (both at Level 2 and Level 3)
DTE/DCE selection
Buffers "allowed" per call

The gateway will have one or more address translation tables for the following mappings:

Calls from X25 to Econet -

Called sub-address to NS destination address

First component of called YBTS address to NS destination address

Calls from Econet to X25 -

First component of called NS address to X25 address

Note that gateway title-to-address conversion tables can be great consumers of memory. For outbound calls, there may be a case for associating specific X25 facilities with the destination address to ease the problem of using the NS to convey QOS parameters which are unknown to it. (It is believed that the Camtec Pad uses this technique).

This information may be hard-coded into the gateway initially, but eventually a flexible system is required to enable the local gateway manager (human being) to change the parameters. This may be accomplished by having the gateway read in a file at system start-up, the file residing at an Econet File Server.

5.2 Monitoring

A statistical report can be requested from the STS module at any time, though the method of initiating this, and recording the information, is to be decided. Without human intervention, the gateway could solicit such a report on a regular basis (say each hour) and file the result away over the Econet. If a gateway manager is 'logged in' to the gateway, he should be able to request such reports and receive the information directly. It appears that the STS can issue unsolicited Report primitives. These can be filed to the file server, or sent to the gateway manager if present:

5.3 Control

There is relatively little that for a gateway manager to do on-line. However he can start and stop the X25 link which would be the cue for the gateway to read in fresh configuration information, if present. The gateway will need to offer an internal X29-like interface for the manager. This means a separate block of code which accepts NS calls over the Econet. and treats the incoming data units as command lines. The protocol used over the NS should be "X29 over NS" as described in previous sections. A future enhancement to this X29 server would allow it to accept X29 calls originating from a remote X25 PAD. This could ease the problems of maintenance and support of the gateway in the field where an engineer's visit would be prohibitively expensive.